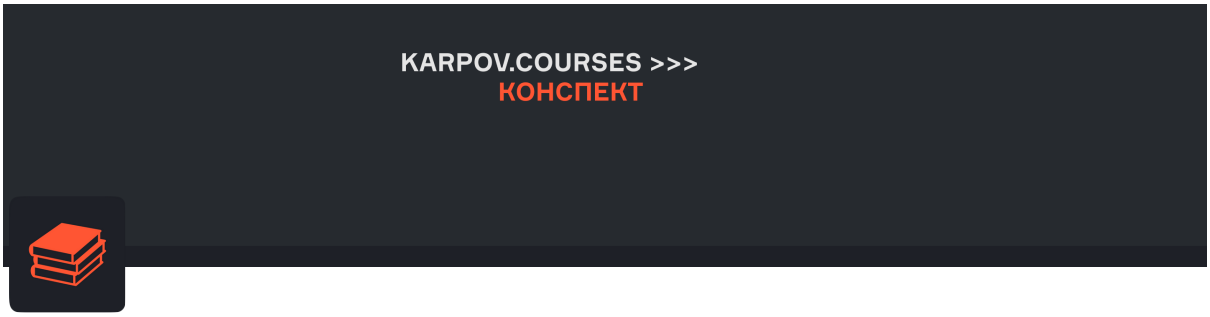


> Конспект > 2 урок > Разбор методов построения Uplift-моделей

> Оглавление

P.S. По ходу лекции лектор регулярно обращается к следующему IPython Notebook. Мы рекомендуем с ним ознакомиться и следовать за лектором, одновременно читая конспект.

> Условные обозначения

Вспомним условные обозначения и термины, используемые в Uplift-моделировании:

X — признаки пользователя: его описание и контекст.

$T \in 0, 1$ — флаг воздействия на пользователя (было ли оно).

$Y(1)$ — целевая переменная во вселенной, где на пользователя было оказано воздействие.

$Y(0)$ — целевая переменная во вселенной, где воздействия не было.

$Y = Y(1)T + Y(0)(1-T)$ — целевая переменная в нашей вселенной.

Один из важных показателей — Individual Tratment Effect (ITE) (также известен как Causal Effect (CE)):

$$ITE = Y(1)-Y(0)$$

Необходимо понимать, что ITE принципиально невозможно наблюдать, потому что невозможно одновременно воздействовать и не воздействовать на клиента.

Был также введён условный средний эффект от воздействия CATE (Conditional Average Treatment Effect):

$$CATE(x) = \tau(x) \overset{def}{=} E[Y(1)|X = x]-E[Y(0) \mid X = x]$$

Или учитывая условия CIA:

$$\tau(x) = E[Y \mid X = x, T = 1]-E[Y \mid X = x, T = 0]$$

Напомним также и о среднем эффекте от воздействия ATE (Average Treatment Effect):

$$ATE(x) \overset{def}{=} E[Y(1)]-E[Y(0)]$$

И при условии независимости T и X получим:

$$ATE(x) = E[Y \mid T = 1]-E[Y \mid T = 0]$$

Для удобства введём обозначение:

$$\mu_t \overset{def}{=} E[Y \mid X = x, T = t] = E[Y(t) \mid X = x]$$

Также напомним и про Propensity Score:

$$e(x) \overset{def}{=} P(T = 1 \mid X = x)$$

Его смысл следующий: если в эксперименте флаг воздействия выбирался случайно (как обычно и происходит) с вероятностью p, то $e(x) = p$

> T-Learner

Первый метод построения Uplift-моделей, который мы разберём называется T-Learner (T от слова two — два). Это довольно простой метод, который заключается в построении отдельной модели для каждой из групп (целевой и

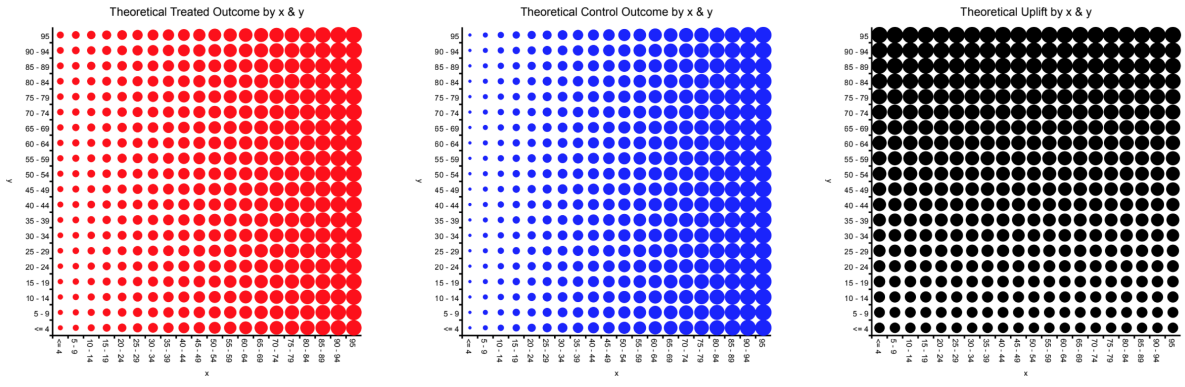
контрольной), которая прогнозирует целевой показатель `Y`. И в качестве прогноза CATE (т.е. по сути Uplift) мы выдаём разность прогнозов этих двух моделей.

Алгоритм следующий:

1. Построим две модели $\hat{\mu}_0(x)$ и $\hat{\mu}_1(x)$, предсказывающие целевой показатель Y: одну для целевой группы (X^1, Y^1), т.е. на которую оказывается воздействие, и вторую для контрольной группы (X^0, Y^0)
2. Спрогнозируем Uplift (CATE) $\hat{\tau}(x)$ как разность $\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x)$

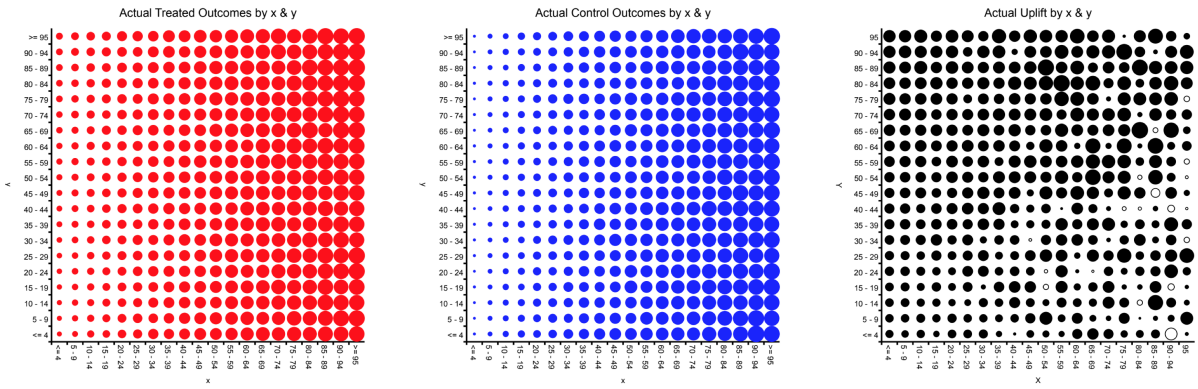
На практике метод T-learner работает хуже других подходов, особенно при небольшом количестве данных. Одна из основных причин: мы предсказываем Uplift не напрямую, а через целевую переменную, поэтому модель обращает внимание на факторы, влияющие больше на сам Y, а не на его прирост.

Это проиллюстрировано на двух примерах ниже. Мы будем предсказывать Uplift от двух факторов: x и y. Видно, что для прогноза этого показателя важен фактор x. В итоге построенные модели будут больше обращать внимание на показатель x . В то же время, если посмотреть на распределение Uplift, для него решающим будет y, а не x.



Размер кружка пропорционален среднему значению целевой переменной (или CATE для 3-го графика) при таких значениях факторов (x, y).

Если теперь довериться полученным моделям и посмотреть спрогнозированный Uplift, то получим странную случайную картину (снизу справа). В целом видно, что алгоритм плохо улавливает зависимость от y.



> X-Learner

Теперь рассмотрим модель X-Learner. Она получила такое название благодаря своей перекрёстной схеме прогноза.

1. Сначала, как и в предыдущей модели, мы построим две модели $\hat{\mu}_0(x)$ и $\hat{\mu}_1(x)$ на контрольной и целевой выборках. В качестве целевых переменных будем использовать Y^0 и Y^1 соответственно.
2. Затем сформируем новые целевые переменные: $\tilde{D}_i^1 = Y_i^1 - \hat{\mu}_0(X_i^1)$ и $\tilde{D}_i^0 = Y_i^0 - \hat{\mu}_1(X_i^0)$ (здесь и появляется перекрёстность). В данном случае мы получим грубую прикидку прироста целевого показателя относительно среднего ожидаемого значения этого показателя в случае, если бы мы не воздействовали на него.
3. Далее уже на новых полученных парах (X^0, D^0) и (X^1, D^1) мы построим модели $\hat{\tau}_0(x)$ и $\hat{\tau}_1(x)$. Фактически эти модели уже близки к тому, что нужно прогнозировать, а именно к Uplift. В целом каждую из них в отдельности можно использовать в качестве прогноза CATE.
4. Итоговый прогноз CATE (Uplift) делается по формуле: $\hat{\tau}(x) = g(x)\hat{\tau}_0(x) + (1-g(x))\hat{\tau}_1(x)$, где коэффициент $g(x)$ зависит от x. Авторы метода советуют выбирать $g(x) = 1$ в случае, когда целевая группа намного больше контрольной, и $g(x) = 0$ наоборот. В более сбалансированных случаях они рекомендуют в качестве $g(x)$ брать модель прогноза propensity score $\hat{e}(x)$. Её можно отдельно строить на данных эксперимента, если эксперимент проводили не вы.

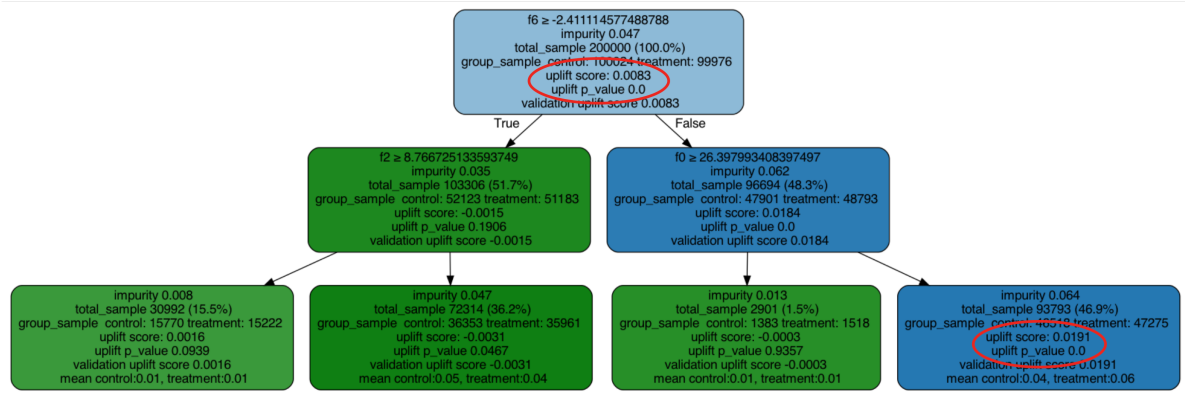
> R-Learner

Следующий подход называется R-Learner, который также относится к группе так называемых мета-подходов. Это группа подходов, под капотом которых можно использовать любые типы моделей машинного обучения, которые будут прогнозировать необходимую величину. Данный подход устроен следующим образом:

1. Разбиваем нашу обучающую выборку на Q фолдов (обычно на 5–10).
2. Для каждого фолда q мы строим две модели: $\widehat{\mu}^{(-q)}(x)$ и $\hat{e}^{(-q)}(x)$. Каждая модель обучается на всех фолдах, кроме выбранного фолда q (отсюда и минус в индексе) $(X^{(-q)}, Y^{(-q)})$ и $(X^{(-q)}, T^{(-q)})$. Как результат будем иметь $2Q$ моделей. Причем, разбиение на фолды не зависит от разбиения на контрольную и целевую группы.
3. Строим модель $\tau(x)$ для минимизации следующей функции потерь: $L_n(\tau(x)) = \frac{1}{n}\sum_{i=1}^{n}\left(\left(Y_i - \widehat{\mu}^{(-q(i))}(X_i)\right) - \left(W_i - \hat{e}^{(-q(i))}(X_i)\right)\tau(X_i)\right)^2$

> Uplift-деревья

Uplift можно также моделировать, используя решающие деревья. Пример простого решающего дерева для Uplift-задачи можно увидеть ниже.



Пример решающего дерева для Uplift-моделирования

Видно, что, как и в обычных решающих деревьях, происходит разбиение всех данных по значению какого-то признака. Это разбиение происходит согласно некоторому критерию. То есть мы хотим разбить данные таким образом, чтобы прогноз Uplift различался максимально в листьях.

Как же оценить Uplift в листьях? Будем просто сравнивать средние значения Y в целевой и контрольной группах:

$$\widehat{\tau}_{\text{node}} = \frac{\sum_{i \in \text{node}} Y_i T_i}{\sum_{i \in \text{node}} T_i} - \frac{\sum_{i \in \text{node}} Y_i \left(1 - T_i\right)}{\sum_{i \in \text{node}} \left(1 - T_i\right)}$$

Разбиение в вершине выбирается по принципу максимизации (оптимизации) некоторого критерия. Самый простой из них — максимизация прироста: $\Delta = \left|\widehat{\tau}_{\text{left}} - \widehat{\tau}_{\text{right}}\right|$

Другими критериями могут служить критерии, основанные на сравнении распределений. Сравнивать распределения можно с помощью так называемых дивергенций (меры различий, которые не являются метриками в полном смысле этого слова, а лишь обладают некоторыми их свойствами).

Пусть на множестве значений $\{y_1, \ldots, y_m\}$ заданы два распределения: $P = \{p_1, \ldots, p_m\}$ и $Q = \{q_1, \ldots, q_m\}$. Тогда можно определить дивергенцию между этими распределениями. Вот некоторые из них:

- Дивергенция Кульбака-Лейблера: $KL(P||Q) = \sum_i p_i \log \frac{p_i}{q_i}$
- Энергетическое расстояние (Energy distance): $E(P||Q) = \sum_i (p_i - q_i)^2$
- Расстояние χ^2 (Пирсона): $\chi^2(P||Q) = \sum_i \frac{(p_i - q_i)^2}{q_i}$

Соответственно, критерий разбиения будет аналогичным:

$$\Delta = D_{aftersplit} - D_{beforesplit}$$

$$D_{beforesplit} = D(P^T_{node} \| P^C_{node})$$

$$D_{aftersplit} = \sum_{child \in left, right} \frac{N(child)}{N(node)} D\left(P^T_{child} \| P^C_{child}\right)$$

Также есть ещё один критерий — Contextual Treatment Selection. Этот метод больше подходит, когда нужно выбирать, какой из методов воздействия применить, но тем не менее он подходит и для случая с единственным воздействием.

В этом подходе мы пытаемся максимизировать следующий показатель:

$$\widehat{\Delta\mu}(s) = \hat{p}\left(\phi_l \mid \phi\right) \times \max_{t=0,\ldots,K} \hat{y}_t\left(\phi_l\right) + \hat{p}\left(\phi_r \mid \phi\right) \times \max_{t=0,\ldots,K} \hat{y}_t\left(\phi_r\right) - \max_{t=0,\ldots,K} \hat{y}_t(\phi)$$

Выглядит громоздко и непонятно, но на самом деле смысл этого выражения в том, что мы разбиваем наши данные на две группы таким образом, что если в каждой из этих групп выбрать тот вариант воздействия, который приносит

наибольшую выгоду, то взвешенная сумма этих максимальных прогнозов будет лучше, чем если мы оставим все записи в одной исходной группе.

> Резюме

В этой лекции мы рассмотрели 3 мета-алгоритма построения Uplift-моделей:

1. T-Learner
2. X-Learner
3. R-Learner

Также мы увидели, что решающие деревья тоже можно использовать, как Uplift-модель, причем есть множество критериев построения таких решающих деревьев, которые основаны на так называемых дивергенциях.

В заключение необходимо отметить некоторые "плюсы" и "минусы" подхода с решающими деревьями:

Плюсы (+):

- Деревья прогнозируют Uplift напрямую — в каждом листе получается несмещённая оценка.
- Легко контролировать надежность алгоритма (робастность).
- Довольно легко и удобно интерпретировать результаты (можно смотреть на структуру дерева).

Минусы (-):

- К сожалению, в текущей реализации они очень медленно строятся.
- CTS кажется несколько “странным” критерием в случае с единственным типом воздействия: он постоянно пытается выделить подвершину, в которой CATE будет отрицательным. Метод будет плохо работать с данными, в которых истинный CATE на всех пользователях положительный.